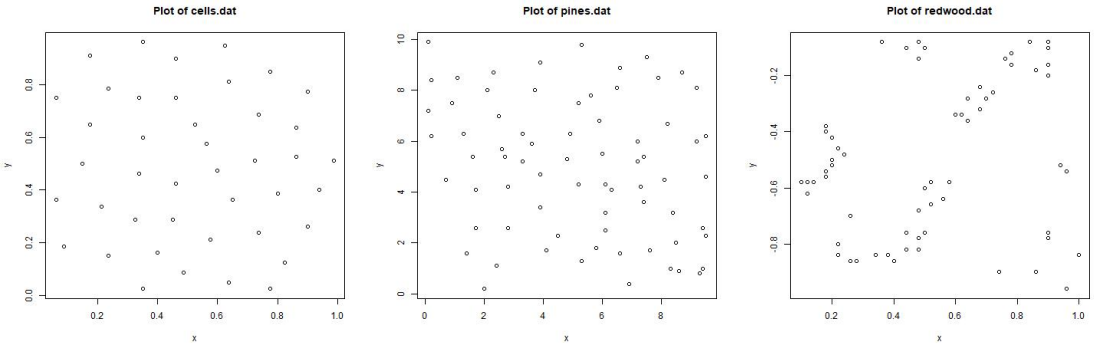
TMA4250: Project 2

Nora Røhnebæk Aasen and Elias Klakken Angelsen

March 20, 2022

Table of Contents



(a) Plot of the cells data (b) Plot of the pines data (c) Plot of the redwood data

Figure 1: Plot of the three data sets we use in problem 1.

1 Problem 1

In this problem we look at three datasets: *cells.dat, pines.dat* and *redwood.dat*.

1.1 a)

First we plot the three datasets as seen in figure 1.

The cell data we see in figure 1a have repulsion traits. This can be explained by the fact that the cells cannot be closer than the radius of the cell body. Such repulsions are hard-core repulsion potentials, ensuring the points have at least a fixed distance from each other.

The redwood trees in figure 1c look like they tend to cluster together, which makes sense as their seeds would have to leave the branches and fall to the ground to become a new tree. This should also be the case for the pine trees, so we can't argue this way. One option is that this could be a snapshot of a young forest starting to emerge from a few "ancestors".

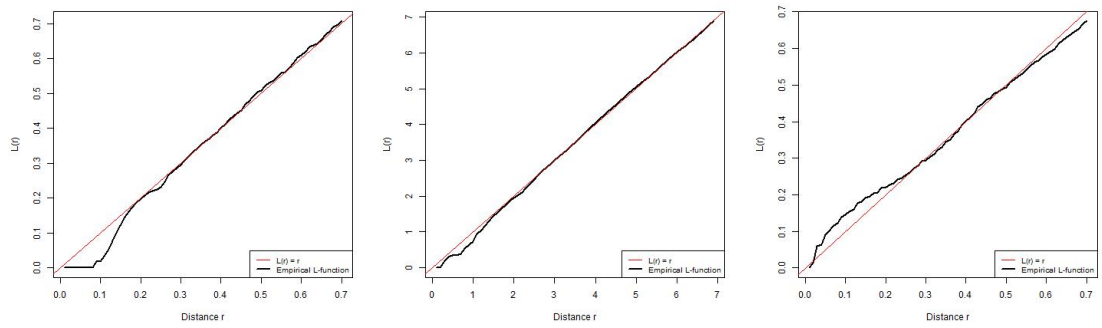
Redwood trees are quite sensitive to soil conditions, and therefore they often depend on making their own biotic community. They are stronger together, as they can nurse the complex soils together, instead of alone, yielding a plausible explanation for the clustering effects we observe.

The pine trees in figure 1b look like they are modeled from either a homogeneous Poisson RF or from a soft repulsion potential model. Both can be true. These trees thrive even on poor soil and can be quite dominant. Therefore, them spreading "all over" is possible. Since they thrive everywhere, a homogeneous Poisson RF makes sense, but the trees could also want to be at "a suitable distance" from their kin, to get more of the resources for themselves. This would mean we have a soft repulsion potential.

1.2 b)

The L-function for a stationary point process N on $\mathbb{R}^d$ is defined as

$$L(r) = \sqrt[d]{K(r)/b_d},$$



(a) Empirical L-function for the cells data

(b) Empirical L-function for the pines data

(c) Empirical L-function for the redwood data

Figure 2: Plot of the L-function for each of the three data sets we use in problem 1.

where b_d is the volume of the unit ball in d dimensions. $K(r)$ is the K-function defined as

$$K(r) = \frac{1}{\lambda} E_0[N(b(0,r) \setminus \{0\})],$$

where E_0 means we are taking the expectation conditioned on us knowing there is a point at 0. λ is the intensity of the point process, describing the mean behavior.

For a homogeneous poisson process on $\mathbb{R}^d$, we can show that

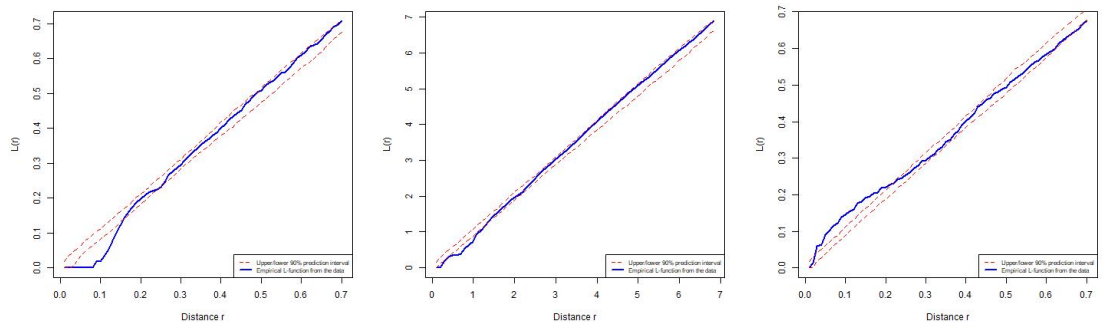
$$K(r) = b_d r^d \implies L(r) = \sqrt[d]{\frac{b_d r^d}{b_d}} = r.$$

If a process has clustering, we expect $L(r) > r$, and if we have repulsion, we expect $L(r) < r$.

We use the `Kfn` function in R to compute the L-function empirically as this spits out $L = \sqrt{K/\pi}$, where K is Ripley's K-function. In $\mathbb{R}^2$, the outputted expression is exactly the expression for the L-function we are after, as the volume of the unit ball in $\mathbb{R}^2$ is exactly π. The results is shown in figure 2.

From figure 2b it looks like a homogeneous Poisson process fits the pine data quite well, even though there is some tendencies towards a slight repulsion at lower distances, as mentioned before. If we consider the redwood data in figure 2c it seems to cluster more at lower distances since we obtain $L(r) > r$. The cell data in figure 2a on the other hand, looks like it yields $L(r) = 0$ for $0 \leq r \leq 0.08$, meaning we probably have a hard-core repulsion potential at these radii, as expected. Both the redwood and cell data tends to act more as a homogeneous Poisson process at larger distances. This may be explained by the fact that cells are allowed to stay close, as long as they avoid "going inside each other". Redwood trees do want the biotic community they build together, but if they go far away from each other, they do not depend on other biotic communities.

As the cell data exhibits hard-core repulsion properties and the redwood tree data exhibits clustering properties at close distances, a homogeneous Poisson model would not be suitable for these data sets, as it does not capture this behaviour. The case for the pine tree data is harder, as this exhibits a weaker repulsion. One can use a homogeneous Poisson model, but that would



(a) L-function for the cells data. (b) L-function for the pines data (c) L-function for the redwood data

Figure 3: Plot of the L-function for each of the three data sets along with a 90% confidence interval calculated from a homogeneous Poisson process conditional on the number of observations in each respective data set.

not take the repulsion into account. A Gibbs model (e.g. Strauss) could be a better alternative, yielding some flexibility in choosing the distance at which the repulsion acts and the strength of the repulsion.

1.3 c)

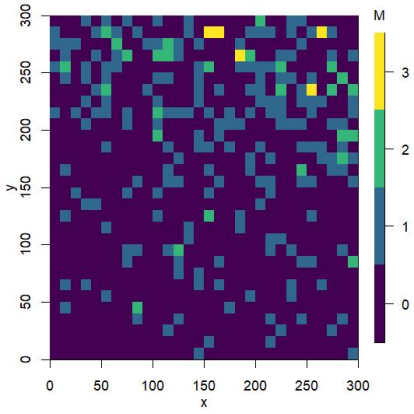
In this problem we wish to compare the L-function for each of our data sets up against the L-function belonging to a homogeneous Poisson process, conditional on the number of observations in each of the data sets. This will give an indication of whether it is reasonable to assume that each of the data sets belong to a homogeneous Poisson process. The results can be seen in figure 3.

As we see in the plots 3a and 3c, the empirical L-function for the cell data and the redwood tree data falls way outside the prediction interval at shorter distances, due to the repulsion and clustering effects, respectively. This is expected. We can observe that the homogeneous Poisson process is not a bad choice of model for long distances, but the misrepresentation at short distances makes it unsuitable for these data sets.

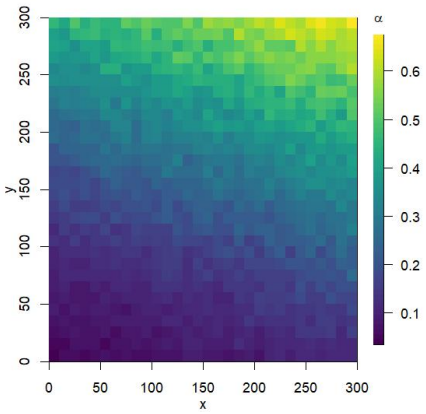
The pine tree data is, as expected, a lot closer to a homogeneous Poisson process, as we can see in figure 3b. Nevertheless, the small anticipated repulsion effect at small distances makes the empirical L-function for the data fall out of the prediction interval we got from the 100 realizations. Even though it looks like a reasonable choice of model for longer distances, we would want the model to represent the weak repulsion potential at the shorter distances, making a homogeneous Poisson model unsuitable for this data set, as well.

2 Problem 2

In this problem we consider a 300×300 square meter sized grid, which there are 900 grid cells within. Within each of these grid cells, an attempt to count the number of pine trees has been made, but there is uncertainty to the accuracy of the counting. Therefore, we also need to consider the detection probability for each grid cell, i.e. how probable were we to observe each individual



(a) The number of observations found in each grid cell.



(b) The detection probability in each grid cell.

Figure 4: Plot of the two data sets we will use in problem 2.

tree in each grid. We denote, for each grid cell $i, j = 1, \ldots, 30$, the counted number of pine trees as $M_{i,j}$, and the detection probability as $\alpha_{i,j}$. We also denote by $N_{i,j}$ the true number of pine trees in each grid cell.

2.1 a)

First we look at a plot of the data, which can be seen in figure 4. We notice that there has been observed more trees in the upper right corner, which makes sense since the detection probability is higher in this area.

We now assume that the observed number of pine trees conditional on the true number of pine trees in each grid cell are independent. Each of our observation $M_{i,j}|N_{i,j}$ then has a binomial distribution with parameters $\alpha_{i,j}$ and $N_{i,j}$. Our joint probability mass function for the full observation model then becomes

$$\boldsymbol{M}|\boldsymbol{N} \sim \prod_{i,j=1}^{30} \binom{N_{i,j}}{M_{i,j}} \alpha_{i,j}^{M_{i,j}} (1-\alpha_{i,j})^{N_{i,j}-M_{i,j}}.$$

2.2 b)

We now assume that the pine trees follow a homogeneous Poisson point process with constant intensity λ. The probability mass function of $\boldsymbol{N}$ would then be

$$p(\boldsymbol{N}) = \prod_{i,j=1}^{30} p(N_{i,j}) = \prod_{i,j=1}^{30} \frac{(\lambda\nu(W))^{N_{i,j}}}{N_{i,j}!} e^{-\lambda\nu(W)},$$

where $\nu(W) = 10^2$ in our case, since each grid cell is 10×10.

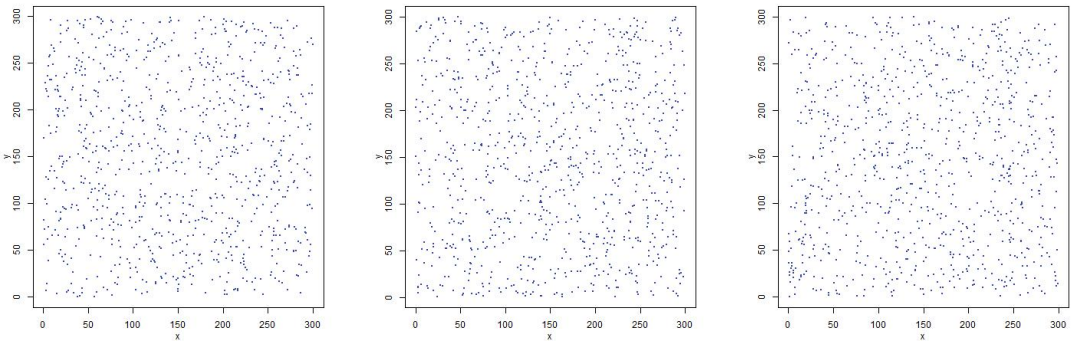


Figure 5: Three simulations of a homogeneous Poisson process on our full grid.

2.3 c)

We know that when $x \sim B(n,p)$, and x and p is know, the MME of n is $n = \bar{X}/p$. We do this grid-wise before averaging to find

$$\hat{\Lambda}_2 = \frac{1}{300^2} \sum_{i,j}^{30} \frac{M_{i,j}}{\alpha_{i,j}}.$$

From our data set, the estimate for $\hat{\lambda} \approx 0.010159$. In figure 5 we plotted three simulations using $\lambda = \hat{\lambda}$.

2.4 d)

We now want to find the probability mass function for $\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}$. For this, we can use Bayes' theorem which gives us that

$$p(\boldsymbol{N}|\boldsymbol{M}) = \frac{p(\boldsymbol{M}|\boldsymbol{N})p(\boldsymbol{N})}{P(\boldsymbol{M})}$$

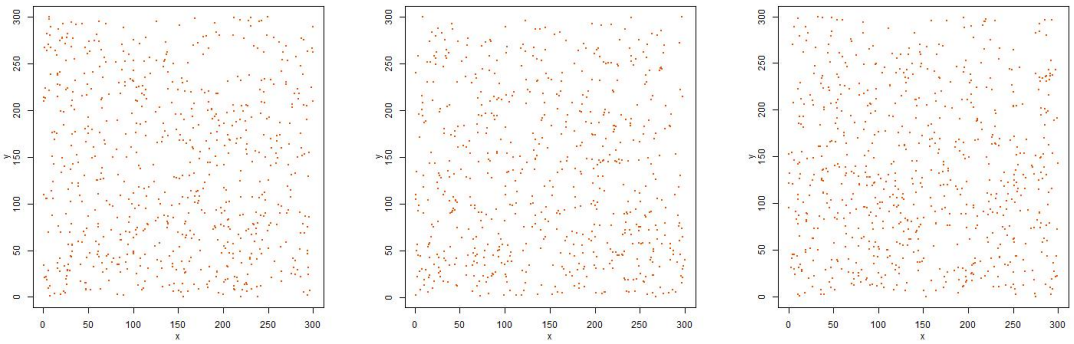


Figure 6: Three simulations of the inhomogeneous Poisson process on our full grid.

From subsection 2.1 and 2.2 we already know the distribution of $p(\boldsymbol{M}|\boldsymbol{N})$ and $p(\boldsymbol{N})$. Furthermore,

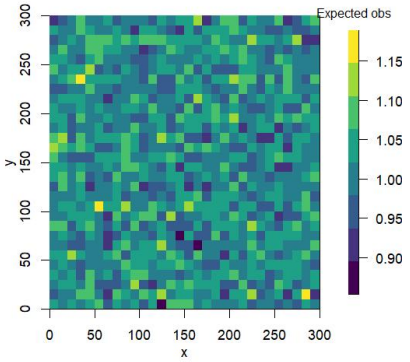
$$\begin{aligned}
p(\boldsymbol{M}) &= \sum_{\boldsymbol{N}\geq\boldsymbol{M}} p(\boldsymbol{M},\boldsymbol{N}) \\
&= \sum_{\boldsymbol{N}\geq\boldsymbol{M}} p(\boldsymbol{M}|\boldsymbol{N})p(\boldsymbol{N}) \\
&= \sum_{\boldsymbol{N}\geq\boldsymbol{M}} \left(\prod_{i,j=1}^{30} \binom{N_{i,j}}{M_{i,j}} \alpha_{i,j}^{M_{i,j}} (1-\alpha_{i,j})^{N_{i,j}-M_{i,j}} \times \prod_{i,j}^{30} \frac{(\lambda\nu(W))^{N_{i,j}}}{N_{i,j}!} e^{-\lambda\nu(W)} \right) \\
&= \left(\prod_{i,j=1}^{30} \frac{\alpha_{i,j}^{M_{i,j}} (\lambda\nu(W))^{M_{i,j}}}{M_{i,j}!} e^{-\lambda\nu(W)} \sum_{\boldsymbol{N}\geq\boldsymbol{M}} \frac{((1-\alpha_{i,j})\lambda\nu(W))^{N_{i,j}-M_{i,j}}}{(N_{i,j}-M_{i,j})!} \right) \\
&= \prod_{i,j=1}^{30} \frac{\alpha_{i,j}^{M_{i,j}} (\lambda\nu(W))^{M_{i,j}}}{M_{i,j}!} e^{-\lambda\nu(W)} e^{(1-\alpha_{i,j})\lambda\nu(W)} \\
&= \prod_{i,j=1}^{30} \frac{(\alpha_{i,j}\lambda\nu(W))^{M_{i,j}}}{M_{i,j}!} e^{-\alpha_{i,j}\lambda\nu(W)}.
\end{aligned}$$

We then get

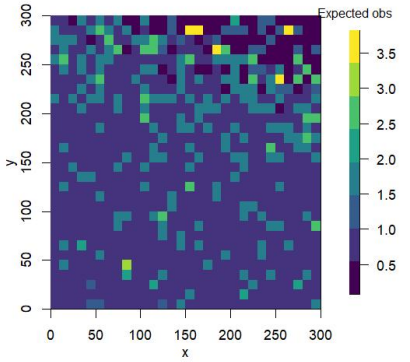
$$\begin{aligned}
p(\boldsymbol{N}|\boldsymbol{M}) &= \prod_{i,j}^{30} \frac{\binom{N_{i,j}}{M_{i,j}} \alpha_{i,j}^{M_{i,j}} (1-\alpha_{i,j})^{N_{i,j}-M_{i,j}} \frac{(\lambda\nu(W))^{N_{i,j}}}{N_{i,j}!} e^{-\lambda\nu(W)}}{\frac{(\alpha_{i,j}\lambda\nu(W))^{M_{i,j}}}{M_{i,j}!} e^{-\alpha_{i,j}\lambda\nu(W)}} \\
&= \prod_{i,j}^{30} \frac{((1-\alpha_{i,j})\lambda\nu(W))^{N_{i,j}-M_{i,j}}}{(N_{i,j}-M_{i,j})!} e^{-(1-\alpha_{i,j})\lambda\nu(W)}.
\end{aligned}$$

We recognize this as an inhomogeneous Poisson process with intensity function $\lambda(s) = (1-\alpha(s))\lambda$.

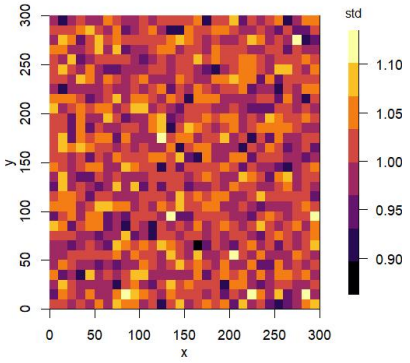
Using this conditional distribution $p(\boldsymbol{N}|\boldsymbol{M})$ with the respective intensity function, we simulate from our inhomogeneous Poisson process. The result is seen in figure 6. It actually seems more spares in the upper right corner where we originally observed the most pine trees. This makes sense, because there were many grids here with high detection probability where we still observed zero trees.



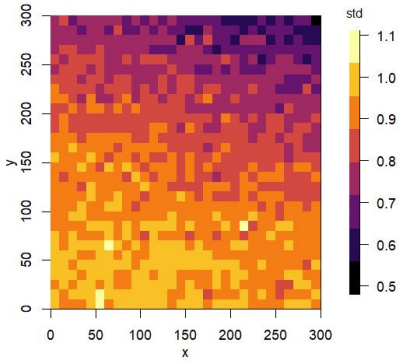
(a) The expected number of observations in each grid based on our prior distribution.



(b) The expected number of observations in each grid based on our posterior distribution.



(c) The standard deviation in each grid based on our prior distribution.



(d) The standard deviation in each grid based on our posterior distribution.

Figure 7: The expected number of points and the standard deviation of the estimates.

2.5 e)

We now want to calculate empirical estimates for $E[\boldsymbol{N}]$ and $E[\boldsymbol{N}|\boldsymbol{M}]$, by calculating the mean after 500 realizations of our prior and posterior distribution. The result, along with the standard deviation, can be seen in figure 7.

Although we weren't able to change the color scale of the plots, we see that the homogeneous Poisson process randomly assigns the expected number of trees to the full area, whereas the inhomogeneous Poisson process has used the information from our data. We see clear similarities between figure 7b and figure 7d to what we saw when looking at the data in figure 4. As expected, the standard deviation is higher in the areas where we had low detection probability.

3 Problem 3

A Neyman-Scott process on $\mathbb{R}^d$, denoted N, is made by generating points (daughter points) from parent points arising from a homogeneous Poisson process. This done in two steps, as we generate the parent points (which do not count as simulated points) from a homogeneous Poisson point process of intensity $\lambda > 0$.

After we have made the parent points, we construct daughter points around each parent point Y. If the sampled number of daughter points around a fixed parent point is denoted C, we generate daughter points independently by defining the i'th daughter point $X_i = Y + R_i r_i$, where R_i is the distance in direction r_i to go from Y. We sample R_i from a pdf giving the distance, and the pdf should be positively truncated to have positive distances. The same should hold for the pdf for the number of points. The direction r_i is sampled from a uniform distribution on the $(d-1)$-sphere S^{d-1}.

Therefore, if the dimension d is given, the parameters we need to specify (in the general case) are the parent intensity $\lambda > 0$, the (nonnegatively truncated) pdf for the number of daughters, often denoted p_C (at least in this course) and the (nonnegatively truncated) pdf for the distance between parent and daughter points, often denoted f_R.

The parent intensity λ can be interpreted as the average number of (parent) points per unit area. It specifies the mean properties by the fact that the expected number of points in a measurable domain B becomes $E[N(B)] = \lambda\nu(B) \cdot E[C]$, where $E[C]$ denotes the expected number of children/daughter points per parent.

Since we assume in this problem that the parent points are distributed from a homogeneous poisson process, we know we need the parent intensity λ_P. The number of points are distributed by a poisson distribution, meaning that we need a "number of daughters"-intensity parameter λ_C, interpreted as the expected number of daughter points per parent point. This means we have the distribution of the number of points per parent $C \sim \text{Poisson}(\lambda_C)$.

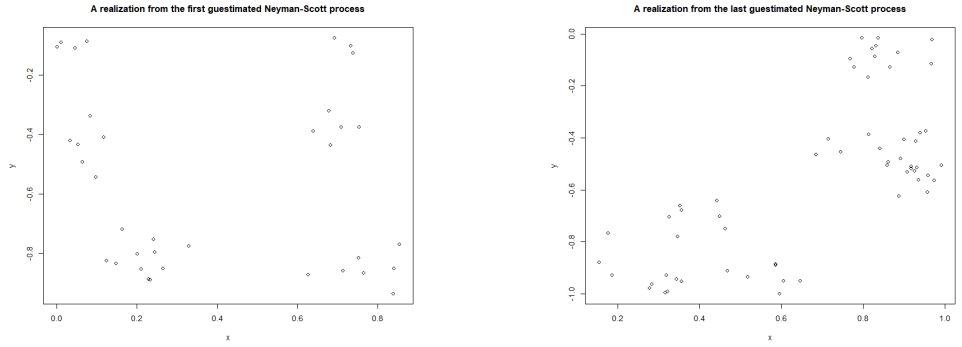
The last pdf to specify is the specification of how the daughter points are distributed around a parent point y. This we know is given by $\mathcal{N}_2(y, \sigma^2 I_2)$ in this problem. This leads us to the parameter σ^2, which describes the variance of the daughter locations. We should expect a symmetric distribution out from the parent point with the parent point in the middle. Since about 40% of the points should end up within one standard deviation from the parent point, we have a way of guessing σ^2 as well.

If we simulate on a finite domain, we know the parent points may land close to the border. This may cause daughter points to land outside of the domain. Or the other way around, if the true parent points lie right outside of the domain, we may only see the daughter points without any reasonable idea on where the parent point lies. It may be tempting to solve this problem by removing the children landing outside of the boundary and generate new children from the same parent point. This yields extra points close to the boundary, as we have restricted the sector in which they may land. One would not want a simulation to act this way.

The easiest way to amend these boundary problems is by extending the process to a window $W \subset \tilde{W}$ such that the actual observation window W works as a "snapshot" of the daughter points in the process.

We now attempt at making an empirical fit of the model parameters to the redwood tree data, where a realization from each attempt can be seen in figure 8.

For the first guestimate, we argue the following way. In the plot of 1c it looks like we can divide



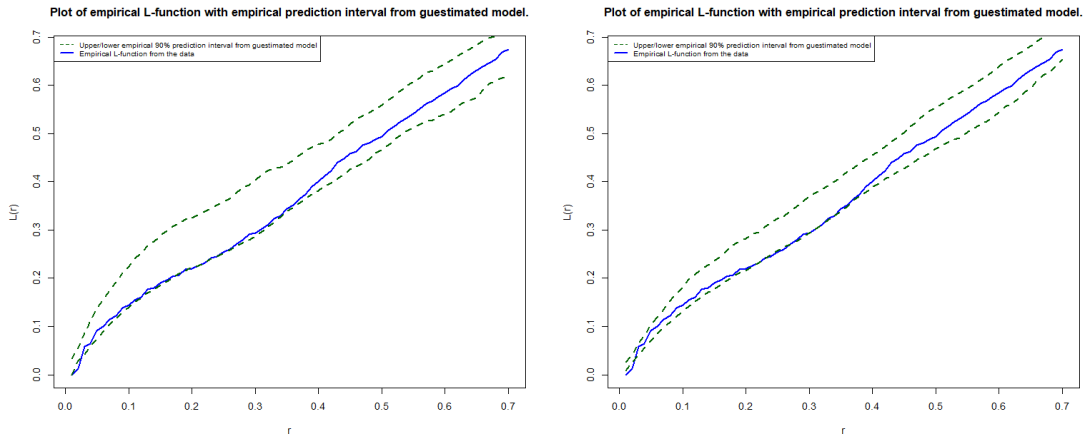
(a) A realization from the (first) guestimated Neyman-Scott model with $\sigma = 0.05, \lambda_p = 9$ and $\lambda_d \approx 6.9$

(b) A realization from the (final) guestimated Neyman-Scott model with $\sigma = 0.05, \lambda_p = 11$ and $\lambda_d = 6$

Figure 8: Two realizations from guestimated Neyman-Scott processes.

this into 9 circular clusters (with some outliers). Since the area of the domain is 1, this should mean the parent intensity can be guestimated to be $\lambda_p = 9$. With these clusters in mind, we can count the number of daughter points and take the mean to find a guestimate for the daughter intensity to be about $\lambda_d = 6.9$.

To guess σ^2, we can use the fact that in two dimensions, around 40% of the points fall into a circle of radius σ. That is, we should have σ to be the "radius" such that the circle of radius σ around the center of each cluster contains about $0.4 \cdot 6.9$ points as sigma. This value is about is about 2.75. Therefore, guessing $\sigma = 0.05$ seems fair, based on how we have guessed the clusters and a rough glance at the plot 1c.



(a) First guess: $\sigma = 0.05, \lambda_p = 9$ and $\lambda_d \approx 6.9$

(b) Last guess: $\sigma = 0.05, \lambda_p = 11$ and $\lambda_d = 6$

Figure 9: Empirical 90% prediction intervals from 100 realizations of the guestimated Neyman-Scott models with the empirical L-function from the redwood data.

This yielded the plot of the L-function as seen in figure 9a. Even though the guestimate is quite

good, we may hope to change the behaviour a bit so that it goes more to the middle of the prediction interval. After testing a bit, we arrived at the parameters $\sigma = 0.05, \lambda_p = 11$ and $\lambda_d = 6$. By keeping the variance fixed, we can make less clustering at lower ranges by decreasing the daughter intensity λ_d, since fewer daughters around each parent yields less clustering effects. This shifts the prediction interval down at shorter distances, yielding a better fit at shorter distances, as we can see in figure 9b.

Since we have 62 points in the original data set, we would want to keep $\mathrm{E}[C]\,\mathrm{E}[P] = \lambda_d\lambda_p$ close to 62. Hence, we changed λ_p to 11, as this gave the better fit than $\lambda_p = 10$.

The final parameters were $\sigma = 0.05, \lambda_p = 11$ and $\lambda_d = 6$.

4 Problem 4

We model the biological cell data as a repulsive point process using the Strauss model. In this model, we have fixed a number of points to work with, denoted n. The pair potential function, $\varphi : [0, \infty) \to \mathbb{R}$ describes the repulsion by using the pair potential function as a repulsive potential energy as a function of the distance between two points.

Our pair potential function φ is given by

$$\phi(r) = \begin{cases} \beta, & r \leq r_0 \\ 0, & r > r_0, \end{cases}$$

where β indicates the strength of the repulsion. To understand this, we consider the pdf of such a process N. By defining the total energy $U : \mathbb{R}^d \times \ldots \times \mathbb{R}^d \to \mathbb{R}$ to be $U(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \varphi\left(\|\vec{x}_i - \vec{x}_j\|\right)$, we can construct a pdf on an observation window W by considering

$$f(x_1, x_2, \ldots, x_n | N(W) = n) = \frac{1}{Z_n} e^{-U(x_1,\ldots,x_n)} = \frac{1}{Z_n} e^{-\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \varphi(\|\vec{x}_i - \vec{x}_j\|)}.$$

Here, Z_n is a unknown normalization constant. This constant is of great importance in statistical physics, and is known as the partition function. In more general cases, it is highly nontrivial to find.

Our repulsion potential tells us that it is harder for points to be closer than a distance r_0 from each other. It is not impossible (unless $\beta = \infty$), but the higher β becomes, the less likely is it for points to cluster together. If two points are further than r_0 away, they do not interact.

Simulating on a bounded observation window W allows points in a repulsion potential to "flee to the boundary", as there are no points outside of W that may repulse them away. Therefore, we often observe more points along the border (especially in corners). This again repels the other points from the border, causing a slight vacuum of points a bit further into the domain. Further in, the points are not as affected by these border issues, and should be observed as following the model.

As with the Neyman-Scott process, we can solve this by expanding the observation window when simulating and rejecting all simulated points that end outside of the wanted observation window. In this problem, we ignore these boundary effects.

We try guestimating parameters to make an empirical model fit to the cell data. Since we work with a fixed number of points (42 points), we only need to guess the radius r_0 and β.

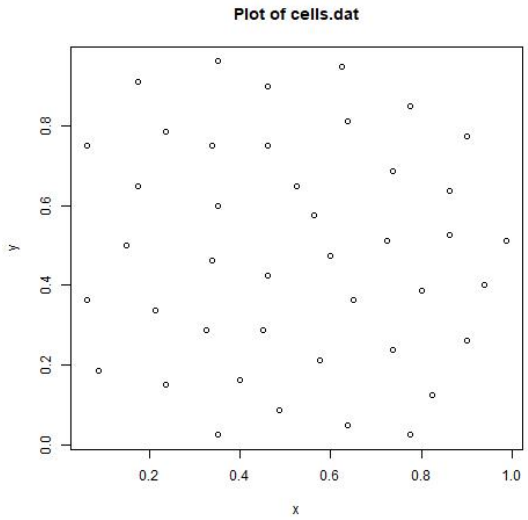
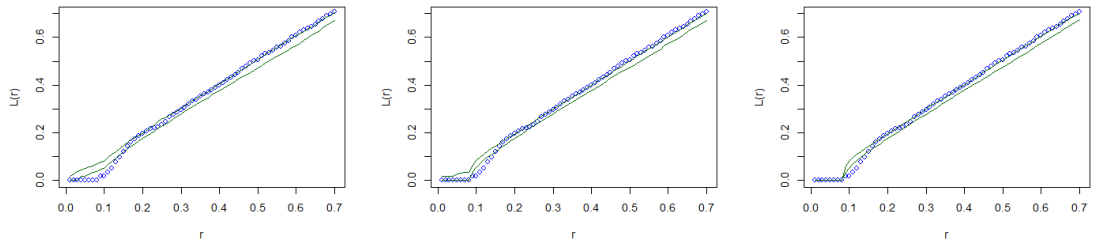


Figure 10: Plot of the cell data set.



(a) First guess: $r_0 = 0.1, \beta = 1$ (b) Second guess: $r_0 = 0.8, \beta = 3$ (c) Third guess: $r_0 = 0.8, \beta = 10$

Figure 11: Plot of the three empirical L-functions from 100 realizations of a Strauss model with guestimated parameters.

As we can see in figure 10, it looks like points stay outside of balls of radius 0.1 from each other, at least most of the time. Therefore, we guess $r_0 = 0.1$ and $\beta = 1$.

We make 100 realizations to find an empirical 90% prediction interval for the L-function. This gives the following plot of the guestimates.

As we can see from the plot in figure 11a, our guestimates gave prediction intervals that suited the model quite ok at longer distances, but at shorter distances, we would need more repulsion. Hence, for the second guestimate, we try $\beta = 3, r_0 = 0.8$. Note that the model does not take into account the extremely week clustering tendencies we see in the L-function for the cell data at long distances.

This change gave the plot we see in figure 11b. It is better, but not perfect. The higher value for β gave us more of the hard repulsion behaviour we want, but not enough at distances in $(0.8, 1.5)$. We try again with $\beta = 10, r = 0.8$ to see if we can fit the model better.

Even though we got a better fit from these, it is far from perfect, as we can see in 11c. The behaviours at small distances (slightly larger than r_0) and at longer distances are not described perfectly. We would probably need a hard-core potential to ensure that points/cells do not crash, and if we could change up the potential function for $r > r_0$ to take the weak clustering effect into account, we could probably get a better fit.

The last parameters we settled on after three guesses were $r_0 = 0.8$ and $\beta = 10$.

We display the cell data with three realizations from the guestimated model.

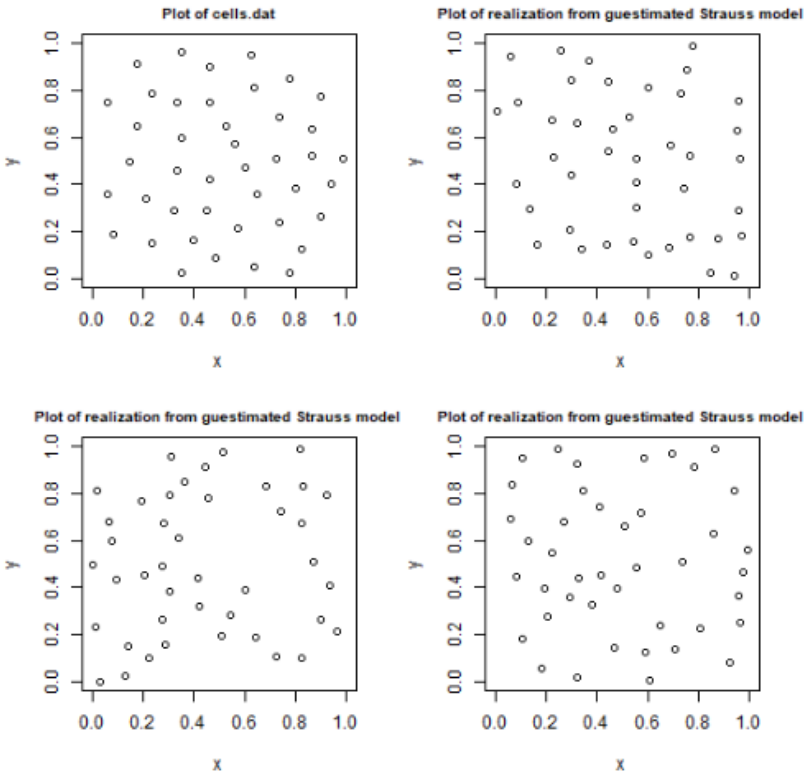


Figure 12: The cell data next to three realizations from our best guestimate.

As we see in 12), the cell data looks more evenly spread out than the three other realizations. The realizations all have some areas towards the middle where there are no points, while the cell data have points evenly spread out.

One thing to note is that the cell data does have points out towards the corners, maybe due to the cells being encapsulated in some body. This may explain why the empirical L-function for the cell data was above the prediction interval we made from 100 realizations, indicating slight clustering, even though the slope was quite similar.

The realizations from the guestimated model utilizes the border more, giving room for some vacuum in the middle, which we do not observe for the cell data. With these visualizations in mind, it would be interesting to see if the guestimated model would fit better if we changed the

observation window to being some circular/elliptic domain a bit outside the outer rim of the cell data.